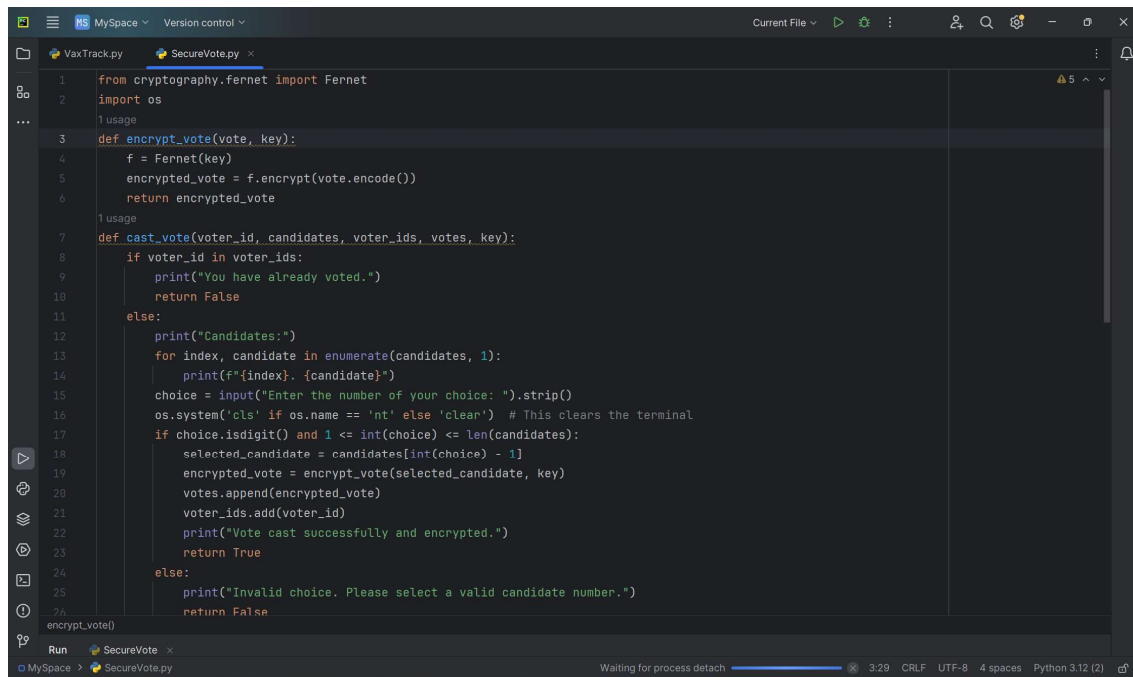
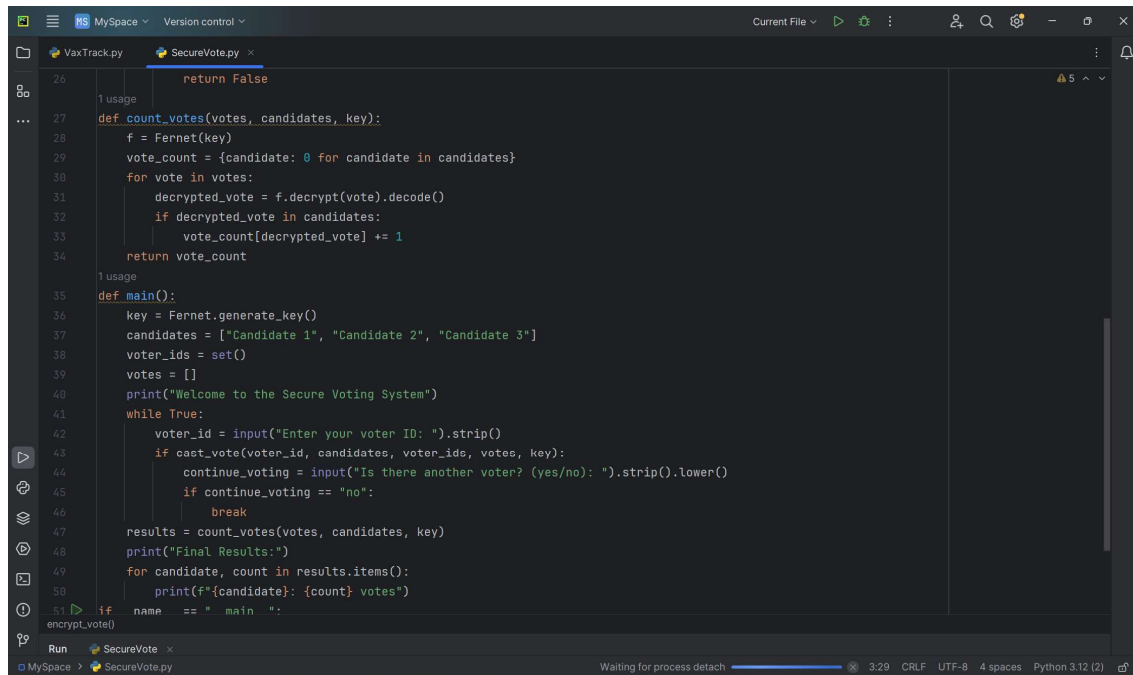


Code Snippets:



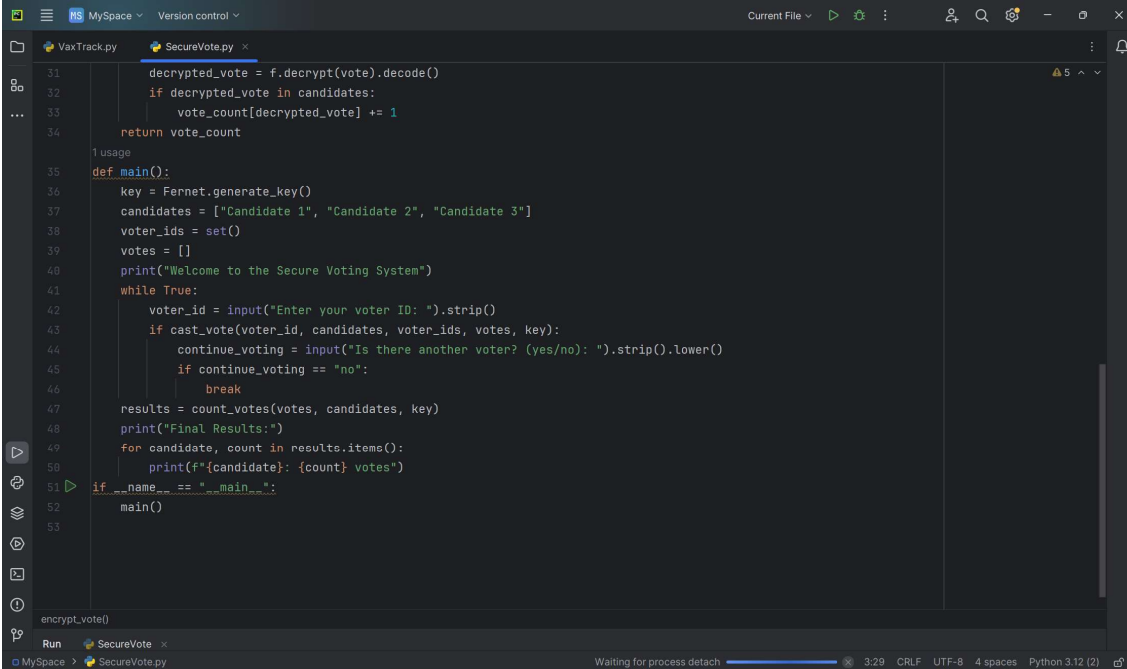
```
1 from cryptography.fernet import Fernet
2 import os
3 usage
4
5 def encrypt_vote(vote, key):
6     f = Fernet(key)
7     encrypted_vote = f.encrypt(vote.encode())
8     return encrypted_vote
9
10 usage
11
12 def cast_vote(voter_id, candidates, voter_ids, votes, key):
13     if voter_id in voter_ids:
14         print("You have already voted.")
15         return False
16     else:
17         print("Candidates:")
18         for index, candidate in enumerate(candidates, 1):
19             print(f"{index}. {candidate}")
20         choice = input("Enter the number of your choice: ").strip()
21         os.system('cls' if os.name == 'nt' else 'clear') # This clears the terminal
22         if choice.isdigit() and 1 <= int(choice) <= len(candidates):
23             selected_candidate = candidates[int(choice) - 1]
24             encrypted_vote = encrypt_vote(selected_candidate, key)
25             votes.append(encrypted_vote)
26             voter_ids.add(voter_id)
27             print("Vote cast successfully and encrypted.")
28             return True
29         else:
30             print("Invalid choice. Please select a valid candidate number.")
31             return False
32
33 encrypt_vote()
```

Run SecureVote x
MySpace > SecureVote.py



```
26 return False
27 usage
28
29 def count_votes(votes, candidates, key):
30     f = Fernet(key)
31     vote_count = {candidate: 0 for candidate in candidates}
32     for vote in votes:
33         decrypted_vote = f.decrypt(vote).decode()
34         if decrypted_vote in candidates:
35             vote_count[decrypted_vote] += 1
36     return vote_count
37
38 usage
39
40 def main():
41     key = Fernet.generate_key()
42     candidates = ["Candidate 1", "Candidate 2", "Candidate 3"]
43     voter_ids = set()
44     votes = []
45     print("Welcome to the Secure Voting System")
46     while True:
47         voter_id = input("Enter your voter ID: ").strip()
48         if cast_vote(voter_id, candidates, voter_ids, votes, key):
49             continue_voting = input("Is there another voter? (yes/no): ").strip().lower()
50             if continue_voting == "no":
51                 break
52         results = count_votes(votes, candidates, key)
53         print("Final Results:")
54         for candidate, count in results.items():
55             print(f"{candidate}: {count} votes")
56
57 if __name__ == "__main__":
58     main()
59
60 encrypt_vote()
```

Run SecureVote x
MySpace > SecureVote.py



```
31     decrypted_vote = f.decrypt(vote).decode()
32     if decrypted_vote in candidates:
33         vote_count[decrypted_vote] += 1
34     return vote_count
35
36 1 usage
37 def main():
38     key = Fernet.generate_key()
39     candidates = ["Candidate 1", "Candidate 2", "Candidate 3"]
40     voter_ids = set()
41     votes = []
42     print("Welcome to the Secure Voting System")
43     while True:
44         voter_id = input("Enter your voter ID: ").strip()
45         if cast_vote(voter_id, candidates, voter_ids, votes, key):
46             continue_voting = input("Is there another voter? (yes/no): ").strip().lower()
47             if continue_voting == "no":
48                 break
49         results = count_votes(votes, candidates, key)
50         print("Final Results:")
51         for candidate, count in results.items():
52             print(f"{candidate}: {count} votes")
53
54 if __name__ == "__main__":
55     main()
56
57 encrypt_vote()
```

How It Works:

1. Start the System:

- You run the program, and it asks for your voter ID.

2. Choose a Candidate:

- The system shows a list of candidates with numbers next to them (e.g., "1. Candidate 1").
- You enter the number of the candidate you want to vote for (e.g., "1" for "Candidate 1").

3. Cast Your Vote:

- The system encrypts your vote to keep it private.
- It clears the screen so no one can see your choice.
- Your vote is recorded, and the system confirms your vote was cast successfully.

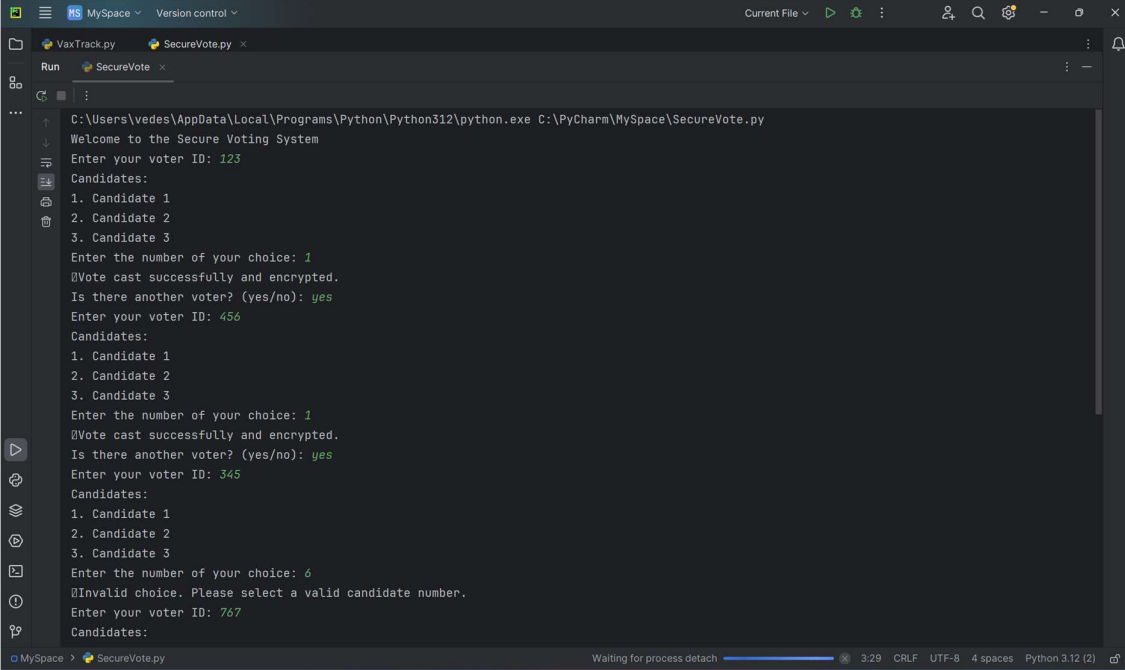
4. Next Voter:

- It asks if there is another voter.
- If yes, it repeats the process. If no, it moves on to the next step.

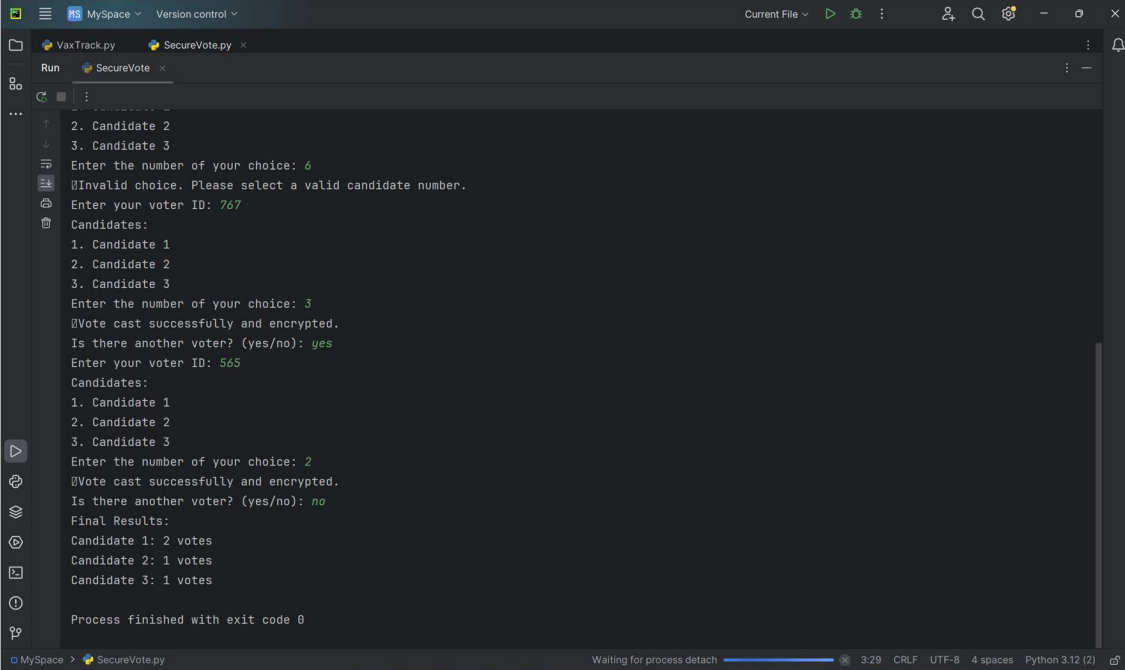
5. Show Results:

- After all votes are cast, the system counts and shows the final results, telling you how many votes each candidate received.

Output:



```
C:\Users\vedes\AppData\Local\Programs\Python\Python312\python.exe C:\PyCharm\MySpace\SecureVote.py
Welcome to the Secure Voting System
Enter your voter ID: 123
Candidates:
1. Candidate 1
2. Candidate 2
3. Candidate 3
Enter the number of your choice: 1
@Vote cast successfully and encrypted.
Is there another voter? (yes/no): yes
Enter your voter ID: 456
Candidates:
1. Candidate 1
2. Candidate 2
3. Candidate 3
Enter the number of your choice: 1
@Vote cast successfully and encrypted.
Is there another voter? (yes/no): yes
Enter your voter ID: 345
Candidates:
1. Candidate 1
2. Candidate 2
3. Candidate 3
Enter the number of your choice: 6
@Invalid choice. Please select a valid candidate number.
Enter your voter ID: 767
Candidates:
```



```
2. Candidate 2
3. Candidate 3
Enter the number of your choice: 6
@Invalid choice. Please select a valid candidate number.
Enter your voter ID: 767
Candidates:
1. Candidate 1
2. Candidate 2
3. Candidate 3
Enter the number of your choice: 3
@Vote cast successfully and encrypted.
Is there another voter? (yes/no): yes
Enter your voter ID: 565
Candidates:
1. Candidate 1
2. Candidate 2
3. Candidate 3
Enter the number of your choice: 2
@Vote cast successfully and encrypted.
Is there another voter? (yes/no): no
Final Results:
Candidate 1: 2 votes
Candidate 2: 1 votes
Candidate 3: 1 votes

Process finished with exit code 0
```